July 2021

GENERATING A SOFTWARE DEFINED SEGMENTATION POLICY FROM STATIC/DYNAMIC ACCESS CONTROL LISTS AND ACTIVE DIRECTORY INTEGRATION

Mnason Pattan

Ankush Arora

Manasi Jain

GENERATING A SOFTWARE DEFINED SEGMENTATION POLICY FROM STATIC/DYNAMIC ACCESS CONTROL LISTS AND ACTIVE DIRECTORY INTEGRATION

AUTHORS:
Mnason Pattan
Ankush Arora
Manasi Jain

ABSTRACT

With multiple customers looking to migrate from traditional campus networks into software defined access (SDA) architectures, it is of paramount importance to enable such transitions in a simplified and automated fashion while still maintaining the current level of segmentation. To address these types of challenges, techniques are presented herein that addresses an important part of such a transition by defining a method of generating a software defined segmentation policy by integrating with an active directory (AD) and referencing the current access policies along with other known methods.

DETAILED DESCRIPTION

Many organizations are transitioning their campuses from a traditional three-tier switching architecture to a software defined access (SDA) architecture. Such organizations are faced with various challenges during a migration. One specific challenge concerns migrating their current access control policies, that are applied at a virtual local area network (VLAN) level (on switched virtual interfaces (SVIs)) or applied dynamically using an identity or policy services engine, to what may be referred to as a software defined segmentation (that employs, possibly among other things, tags (that are, for example, assigned to a user's or device's traffic at ingress) and access policies comprising logical groupings), which decouples access entitlements from specific artifacts such as Internet Protocol (IP) addresses, VLANs, etc.

Building a software defined segmentation policy on an SDA is strategic step for enterprise customers who are looking to achieve end-to-end security with a unified tag-based policy across their SDA, software-defined wide area network (SD-WAN), remote

access virtual private network (VPN), and application centric infrastructure (either on-premise or in a cloud).

Aspects of the techniques presented herein address various of the challenges that were described above, including:

- The current methods that are available for migrating security policies from access control lists (ACLs) are focused on generating IP or application-based policies as output and not a tag-based software defined segmentation policy.
- Methods are lacking to group the static and dynamic ACLs based on their traffic flows, which would be an important step towards creating a software defined segmentation policy.
- There is also no way today to refer to active directory (AD) audit logs and classify and group source user groups, which would become a building block for a software defined segmentation policy.

To construct a software defined segmentation policy the key required components include, for example:

- A source classification to assign source security or scalar group tags.
- A destination classification to assign destination security or scalar group tags.
- Contracts which would limit the ports and protocols that are allowed between a source to a destination and vice versa.
- The action(s) that are to be taken, permitted, or denied.
- The point of enforcement which would depend upon the type of communication (e.g., east-to-west or north-to-south).

To address these types of challenges, techniques are presented herein that support novel methods for source and destination classification as well as a method to identify the point of enforcement. These methods help enable the construction of an accurate and complete software defined segmentation policy before an SDA migration.

An overall high-level diagram of aspects of the techniques presented herein is presented in Figure 1, below.

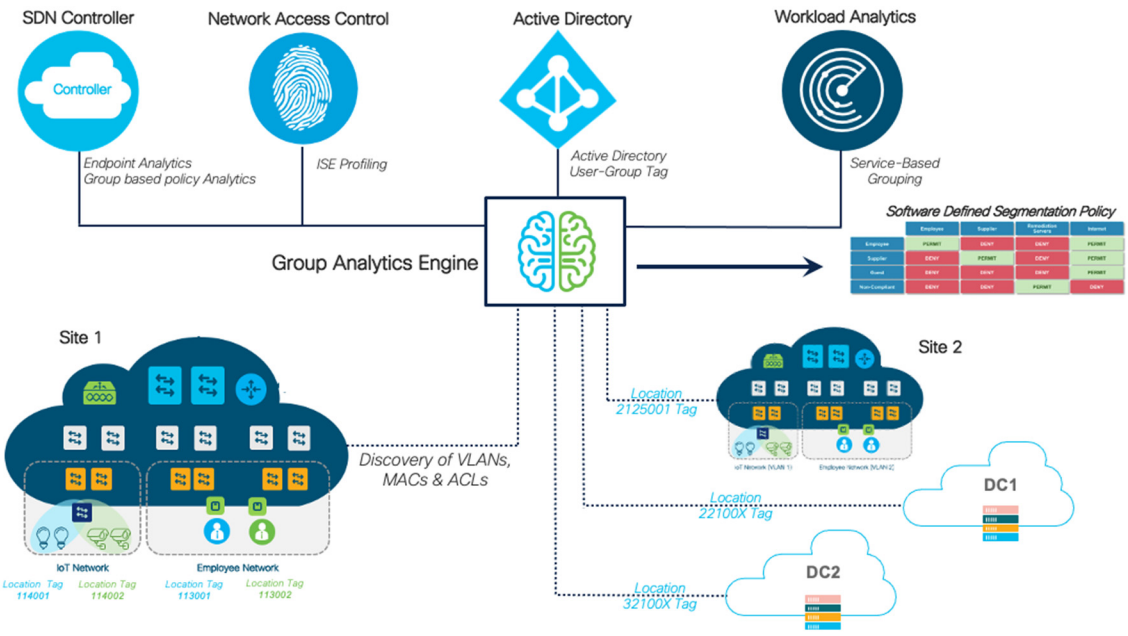

Figure 1: Exemplary High-Level Architecture

The techniques presented herein may be further explored by considering a traditional campus network with some level of segmentation using VLANs and policy controls enforced using static ACLs, as illustrated in Figure 2, below.

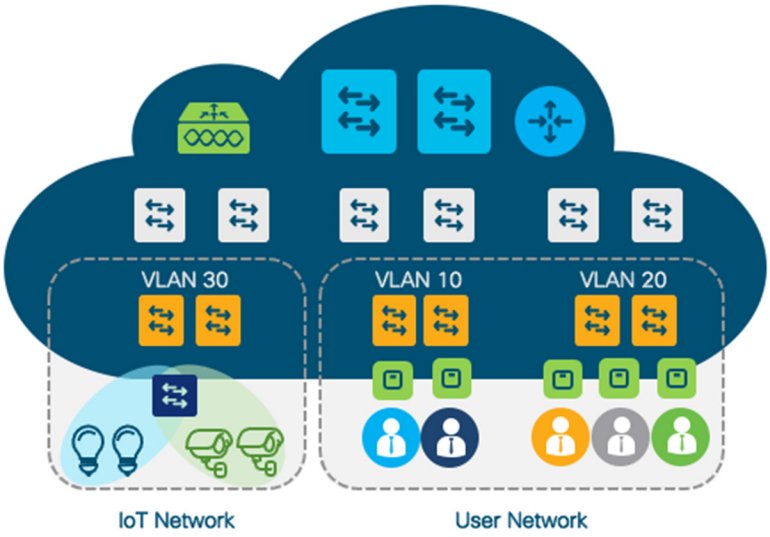

Figure 2: Exemplary Campus Network

As depicted in Figure 2, above, the current state that is being considered is a user network made of two different VLANs (i.e., 10 and 20) with five different user types, as summarized in Table 1, below.

User Type	VLAN	IP Subnet
General Users	10	10.10.10.0/24
Contractors	10	10.10.10.0/24
Function A Users	20	10.10.20.0/24
Function B Users	20	10.10.20.0/24
Traveling Function A Users	20	10.10.20.0/24

Table 1: Illustrative User Network State

Both of the VLANs (i.e., 10 and 20) have static ACLs defined on the VLAN interfaces permitting only specific resources.

The activities of the group analytics engine to classify and derive the various components of a software defined segmentation policy are described in the narrative below in two broad sections.

Under a first section, for classifying the sources, a first step comprises utilizing a known method to discover the endpoint types in a site and within each VLAN. This may be accomplished using identity services profiling or endpoint analytics if a Software Defined Network (SDN) controller is already in place. For scenarios where an SDN controller is not deployed, this may be accomplished by using media access control (MAC) vendor-based profiling.

Following the above classifying activity, the discovery would point to VLAN 30 containing cameras and smart bulbs while VLAN 10 and VLAN 20 would be discovered as VLANs containing workstations.

A next step, which is novel in the techniques presented herein, involves integrating with an AD to assign user group tags and then use data modelling to derive the best possible tags for the sources that are of endpoint type ‘workstation.’ One possible algorithm for carrying out the above-described step is presented in Figure 3, below.

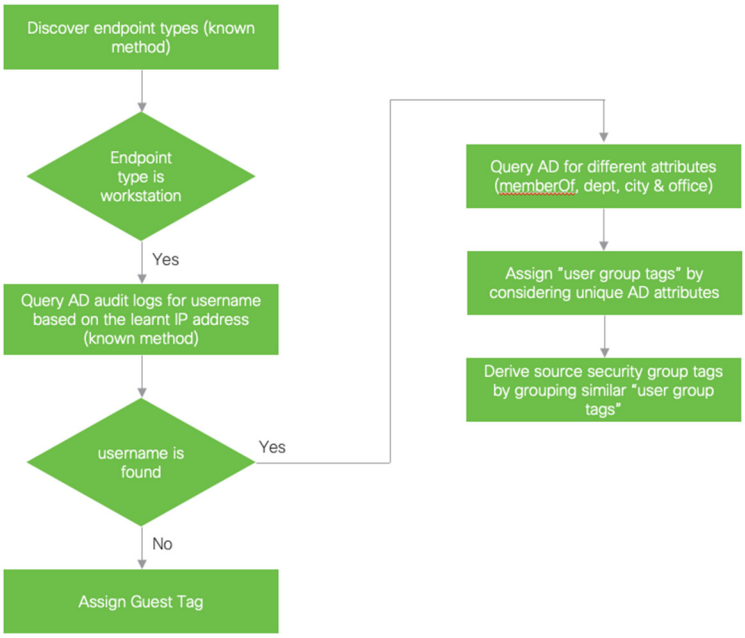

Figure 3: Active Directory Integration Algorithm

The security groups in an AD would not have a one-to-one mapping with the user group tags. The user group tag would be based on a combination of the user's security group membership, location, and department.

Using the same example from above, for VLANs 10 and 20, which are known to contain workstations the active IP addresses, would be fetched from an Address Resolution Protocol (ARP) table. Then, using a known method to read the AD audit logs, an IP-user mapping database may be built as illustrated in Table 2, below.

IP (From ARP)	Username from audit logs
10.10.20.2	User1
10.10.20.3	User2
10.10.20.4	User3
10.10.10.2	User4
10.10.10.3	User5
10.10.20.5	User6
10.10.20.6	User7
10.10.20.7	User8
10.10.10.4	User9
10.10.10.5	User10

Table 2: Illustrative IP-User Mapping Database

Next, the AD may be queried to construct a database of attributes as illustrated in Table 3, below.

User	memberOf	l(City)	physicalDeliveryOfficeName	department
User1	FunctionA Users Domain Users AppA Users AllEmployees AllManagers	Bangalore	Location1	FunctionA
User2	FunctionB Users Domain Users AppB Users AllEmployees	Bangalore	Location1	FunctionB
User3	FunctionA Users Domain Users AllEmployees	Mumbai	Location2	FunctionA
User4	Domain Users AllEmployees Dcloud Users	Bangalore	Location1	FunctionC
User5	Contractors Domain Users	Bangalore	Location1	FunctionC
User6	FunctionA Users Domain Users AppA Users AllEmployees Dcloud Users	Bangalore	Location1	FunctionA
User7	FunctionB Users Domain Users AppB Users	Bangalore	Location1	FunctionB

	AllEmployees			
User8	FunctionA Users Domain Users AllEmployees Dcloud Users	Mumbai	Location2	FunctionA
User9	Domain Users AllEmployees	Bangalore	Location1	FunctionC
User10	Contractors Domain Users	Bangalore	Location1	FunctionC

Table 3: Illustrative Attribute Database

The group analytics engine may then assign the user group tags to the users based on uniqueness while also carrying a pattern for similar attributes. In the instant example the user group tags that are assigned are illustrated in Table 4, below.

User	User-Group Tag	Methodology
User1	001001001001 002001001001 003001001001 004001001001 005001001001	
User2	006001001002 002001001002 007001001002 004001001002	
User3	001002002001 002002002001 004002002001	
User4	002001001003 004001001003 008001001003	
User5	009001001003 002001001003	
User6	001001001001	First three digits represents the memberOf attribute

	002001001001 003001001001 004001001001 008001001001	Next three digits represents the city attribute Next three digit represents the office attribute Next three digit represents the department attribute With each unique attribute that is seen, the counter would be incremented by 1
User7	006001001002 002001001002 007001001002 004001001002	
User8	001002002001 002002002001 004002002001 008002002001	
User9	002001001003 004001001003	
User10	009001001003 002001001003	

Table 4: Illustrative User Group Tags

Under aspects of the techniques presented herein, an algorithm that may be executed for each user for the user group tag assignment is depicted in Figure 4, below, and assumes that only one value for each attribute is currently stored in the database.

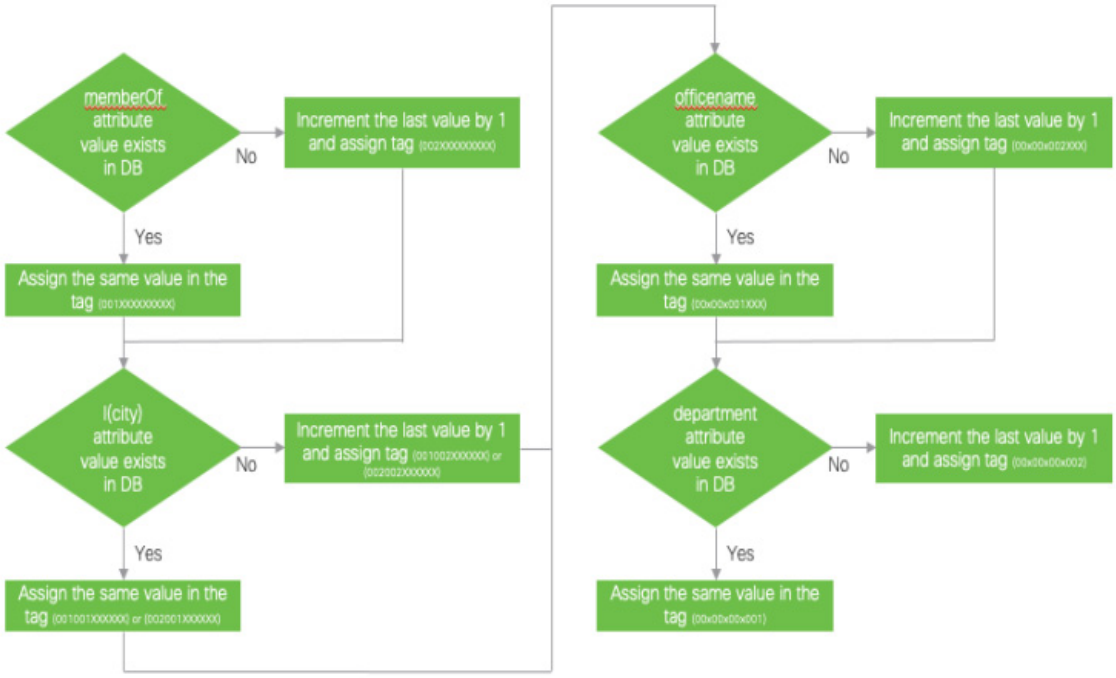

Figure 4: Tag Assignment Algorithm

The group analytics engine may then group based on, for example, unique attributes pattern match and the generic common attributes to recommend the source classification security tags. Based on a user configurable slider to balance the control of granularity versus scale the source security tags may be automatically derived.

The grouping mechanism to derive a security tag from the user group tags is depicted in Figure 5, below.

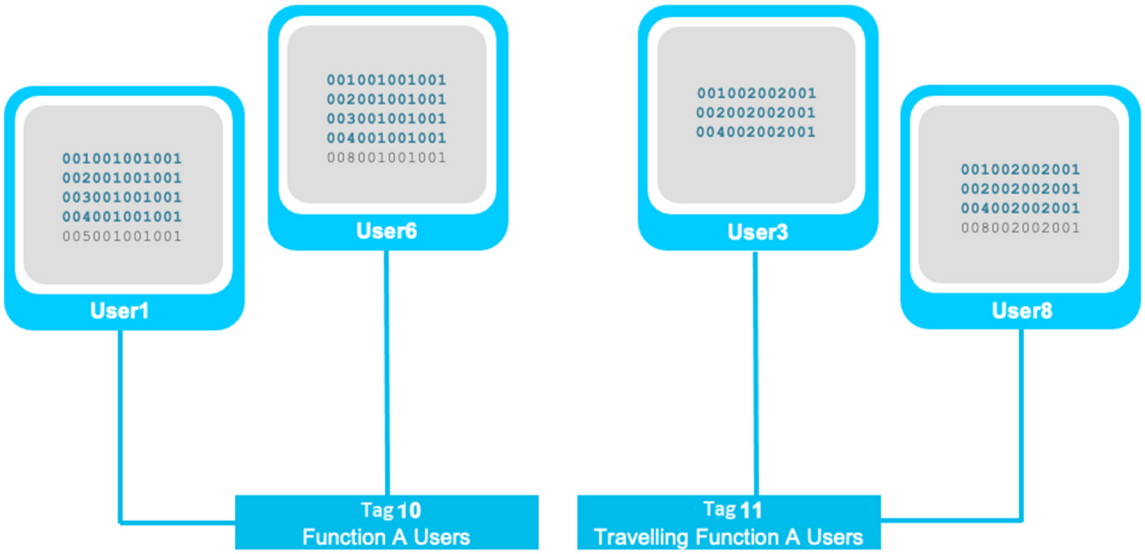

Figure 5: Illustrative Grouping Mechanism

As illustrated in Figure 5, above, based on common user group tags across users different users may be classified as being part of similar functional and location groups and assigned tags.

The group analytics engine may then list the granular to the most generic groupings (as illustrated in Table 5, below) to enable a user to select the right level of granularity while balancing scale.

	Users	Tag	Common AD Groups for policy
Granular	User1 User6	10	FunctionA Users AppA Users
	User3 User8	11	FunctionA Users
	User2 User7	12	FunctionB Users AppB Users
	User4 User9	13	AllEmployees
	User5 User10	14	Contractors
Generic	User1 User2 User3 User4 User6 User7 User8 User9	15	AllEmployees
	All users (1 to 10)	16	Domain Users

Table 5: Illustrative Granularity/Generic Groupings

The granular classifications would consider all of the attributes that were fetched from the AD to identify unique patterns while the generic classifications would consider only the common group memberships to recommend. Such a method provides the flexibility for defining granular and generic catch-all policies in an identity services engine and SDN controller.

Under a second section, for classifying the destinations based on existing ACLs a first step comprises classifying the destinations based on their locations as it applies to a software defined segmentation architecture. The destinations may be discovered and assigned a location group tag with further grouping within the location based on services that are served by the destination as required. The final groups would be assigned a destination security tag.

The location group tag will assign tags for destinations that would map to Data Center (DC) resources, local shared resources, local user groups, local endpoint groups, branch X user groups, etc. These location group tags would be assigned at each site and would be unique across the enterprise. The mapping of the destination IPs found in an ACL to a location group tag would be based on discovery data from all the networks. Such discovery data may be from an IP database (IPDB) of an organization or from a known collector source such as a network traffic flow monitoring facility.

Using known deduplication methods, similar location group tags (e.g., DC resources) may be merged across the different sites to then be assigned with a destination security group tag.

Under aspects of the techniques presented herein, an algorithm that may be executed to accomplish the activities that were described above is presented in Figure 6, below.

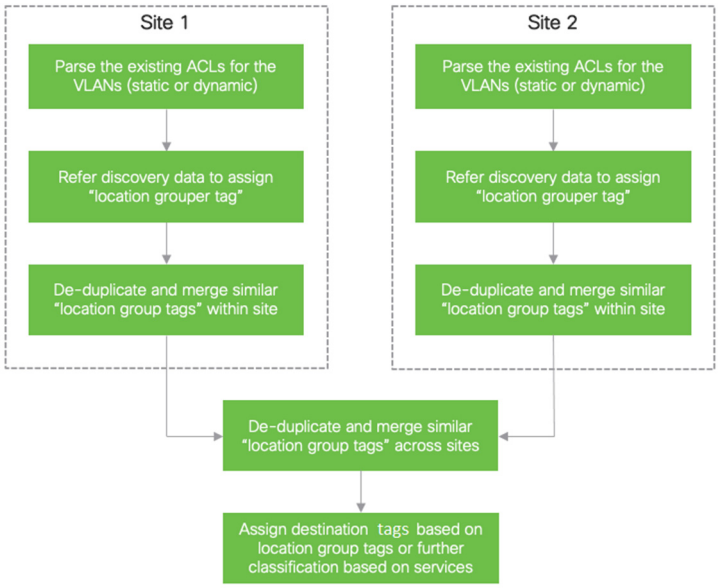

Figure 6: Destination Security Tag Assignment Algorithm

The techniques presented herein may be further explored by considering traditional campus site 1, from Figure 1, above, which has connectivity to different branch sites and DCs as illustrated in Figure 7, below. The users and the endpoints in this site would access the resources in these destinations.

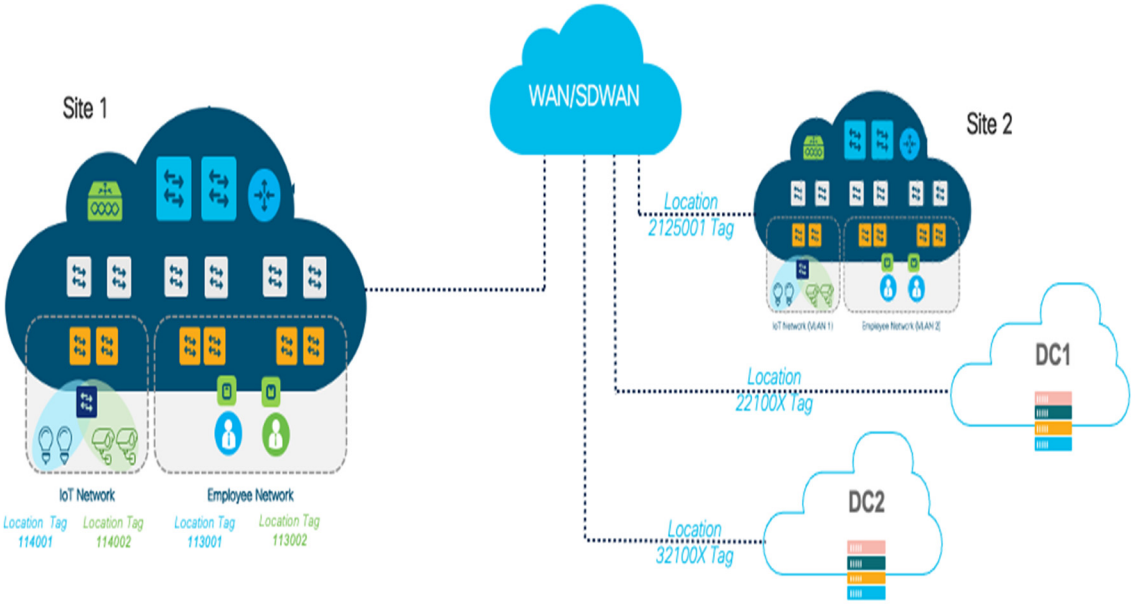

Figure 7: Exemplary Extended Network

Assume that the discovery data that is presented in Table 6, below, is built by an analytics engine, either referring to an IPDB or by discovering the IP interfaces in each location.

IP Network	Location	Discovered Endpoint Types
10.10.10.0/24	Site1	Workstation
10.10.20.0/24	Site1	Workstation
10.10.30.0/24	Site1	Cameras Smart Bulbs
10.10.40.0/24	Site1	Servers/Hypervisors
10.20.10.0/24	Site2	Workstation
10.20.20.0/24	Site2	Workstation
10.20.30.0/24	Site2	Cameras IP Phones
10.30.10.0/24	DC1	Servers/Hypervisors
10.30.20.0/24	DC1	Servers/Hypervisors
10.40.10.0/24	DC2	Servers/Hypervisors
10.40.20.0/24	DC2	Servers/Hypervisors

Table 6: Exemplary Discovery Data

The discovered endpoint types, as presented in Table 6, above, may be based on a known method such as using identity services profiling or endpoint analytics if an SDA is already established. For scenarios where an SDA is not established, this may be accomplished by using MAC vendor-based profiling.

Next, the ACLs may be parsed and location group tags may be assigned as it applies to a software defined segmentation architecture for Site 1. Consider the access list that is applied on VLAN 20's interface as depicted in Figure 8, below.

```
ip access-list extended ACL-ALL-USERS
 remark --== symantec updates ==--
 permit tcp host 10.30.10.10 eq 8014 any
 permit tcp host 10.30.10.11 eq 8014 any
 permit tcp host 10.30.20.10 eq 8014 any
 permit tcp host 10.30.20.11 eq 8014 any
 permit tcp any host 10.30.10.10 eq 8014
 permit tcp any host 10.30.10.11 eq 8014
 permit tcp any host 10.30.20.10 eq 8014
 permit tcp any host 10.30.20.11 eq 8014
 remark --== local DNS ==--
 permit udp host 10.10.40.10 eq 53 any
 permit udp host 10.10.40.11 eq 53 any
 permit udp any host 10.10.40.10 eq 53
 permit udp any host 10.10.40.11 eq 53
 remark --== local user groups ==--
 permit ip any 10.10.10.0 0.0.0.255
 remark --== local endpoint groups ==--
 permit ip any 10.10.30.0 0.0.0.255
 remark --== remote user groups ==--
 permit ip any 10.20.10.0 0.0.0.255
```

Figure 8: Exemplary Access List

The model of policy would be derived as a ‘whitelist model’ and then the group analytics engine would assign ‘location group tags’ as applicable for a software defined segmentation policy as depicted in Table 7, below.

IP	Location-Group Tag	Service	Classification
10.30.10.10	221001	8014	DC1 Service Resources
10.30.10.11	221002	8014	DC1 Service Resources
10.30.20.10	221003	8014	DC1 Service Resources
10.30.20.11	221004	8014	DC1 Service Resources
10.10.40.10	122001	53	Local service resources
10.10.40.11	122002	53	Local service resources
10.10.10.0/24	113001	Any	Local user group
10.10.30.0/24	114001	Any	Local endpoint group
10.20.10.0/24	2125001	Any	Remote1 user group

Table 7: Illustrative Policy Assignments

Referring to Table 7, above, the first digit of a tag indicates whether the IP or subnet is local or remote to the site – i.e., 1 for local, 2 for DC1, 3 for DC2, and 21 for Remote1. The second digit of a tag indicates whether the resource is within the fabric or outside the fabric – i.e., 1 for within fabric and 2 for outside fabric. The third digit of a tag indicates the classification based on identified endpoint types – i.e., 1 for DC-based service resources, 2 for local service resources, 3 for local users, 4 for local endpoints, and 5 for remote users.

The group analytics engine may process the groups with similar resources to classify them with the same destination security tag as depicted in Table 8, below.

IP	Location-Group Tag	Security Tag
10.30.10.10	221001	21
10.30.10.11	221002	
10.30.20.10	221003	
10.30.20.11	221004	
10.10.40.10	122001	22
10.10.40.11	122002	
10.10.10.0/24	113001	23
10.10.30.0/24	114001	24
10.20.10.0/24	225001	25

Table 8: Illustrative Tag Assignments

More granular classifications may be carried out by considering the services while balancing with the scale. Additional known methods may be used to group destinations based on, for example, services in the DCs.

For deriving contracts from ACLs, the static ACLs may be used as reference to derive the contracts between the source and destination. For use cases where the static ACLs do not have services and are generic 'permit IP any, the group-based policy analytics application of a SDN controller may be used to identify and derive contracts. For example, network management ports and protocols may be grouped together to be assigned with a contract, referred to as a Network Management System (NMS) contract.

Based on the 'location group tag' the enforcement point can be identified since the location group tag differentiates between north-to-south (i.e., outside the fabric) and east-to-west (i.e., within the fabric) traffic. This would enable IP-tag mappings to be created only for location groups that are a part of the north-to-south traffic flow.

Figure 9, below, depicts aspects of a derived software defined segmentation policy (based on the level of granularity versus scalability as selected by a customer) where a granular approach was selected for VLAN 20.

	Tag10	Tag11	Tag12	Tag21	Tag22	Tag23	Tag24	Tag25
Tag10	Permit any	Permit any	Permit any	contract	contract	Permit any	Permit any	Permit any
Tag11	Permit any	Permit any	Permit any	contract	contract	Permit any	Permit any	Permit any
Tag12	Permit any	Permit any	Permit any	contract	contract	Permit any	Permit any	Permit any
Tag21	contract	contract	contract	Permit any	Deny any	Deny any	Deny any	Deny any
Tag22	contract	contract	contract	Deny any	Permit any	Deny any	Deny any	Deny any
Tag23	Permit any	Permit any	Permit any	Deny any	Deny any	Permit any	Deny any	Deny any
Tag24	Permit any	Permit any	Permit any	Deny any	Deny any	Deny any	Permit any	Deny any
Tag25	Permit any	Permit any	Permit any	Deny any	Deny any	Deny any	Deny any	Permit any

Figure 9: Illustrative Security Policy

Of particular interest and note in the techniques presented herein as described and illustrated in the above narrative are, for example:

- The use of AD user attributes such as security group membership, location, and department collectively to derive source security tags.
- The use of existing static and dynamic ACLs, discovery data, and novel location-based grouping to derive destination security tags.

- The use of location group tags to identify enforcement points to carry over the current security policy and access control.

In summary, as organizations look to migrate from traditional campus networks into SDA architectures it is of paramount importance to enable those transitions in a simplified and automated fashion while still maintaining the current level of segmentation. Techniques have been presented that address an important part of such a transition by defining a method of generating a software defined segmentation policy by integrating with an active directory and referencing the current access policies along with other known methods.